

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-265459  
(43)Date of publication of application : 07.10.1997

(51)Int.Cl. G06F 15/16

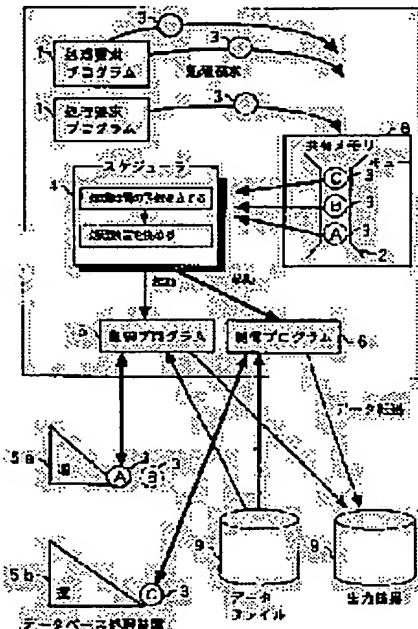
(21)Application number : 08-074340 (71)Applicant : MITSUBISHI ELECTRIC CORP  
(22)Date of filing : 28.03.1996 (72)Inventor : SHIMIZU HIDEHIRO

(54) METHOD FOR CONTROLLING DATA PROCESSOR

**(57)Abstract:**

**PROBLEM TO BE SOLVED:** To execute scheduling capable of suppressing the reversion between the generation order of respective processing requests and the end order of processing and shortening the processing time of the whole information processing system provided with plural data processors having respectively different processing speeds.

**SOLUTION:** In scheduler processing, the estimated processing time of each database processor 5 (5a or 5b) for a generated processing request, the processing time of the whole queue for the processing request in each processor 5 and the remaining time of the processing request being processed by each processor 5 are respectively estimated. Then a processing request generated in the queue of the processor 5 having the shortest estimated processing end time obtained by totalizing the remaining time, the processing time of the whole queue and the estimated processing time is added.



## **LEGAL STATUS**

[Date of request for examination] 23.03.2001

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration] abandonment

[Date of final disposal for application] 06.05.2004

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平9-265459

(43)公開日 平成9年(1997)10月7日

(51)Int.Cl.<sup>o</sup>  
G 0 6 F 15/16

識別記号  
3 8 0

序内整理番号  
F I  
G 0 6 F 15/16

技術表示箇所  
3 8 0 Z

審査請求 未請求 請求項の数6 OL (全13頁)

(21)出願番号 特願平8-74340

(22)出願日 平成8年(1996)3月28日

(71)出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72)発明者 清水 英弘

東京都千代田区丸の内二丁目2番3号 三菱電機株式会社内

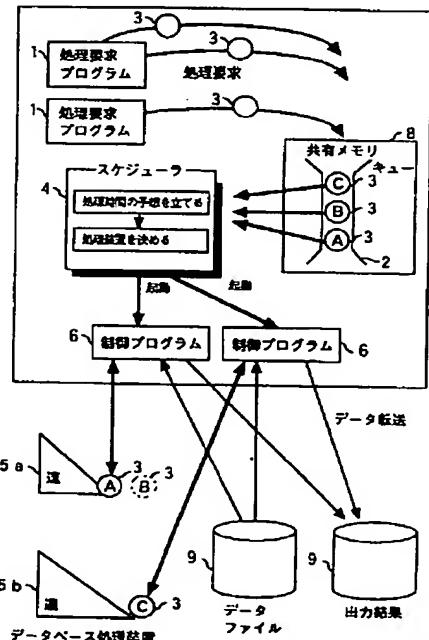
(74)代理人 弁理士 宮田 金雄 (外3名)

(54)【発明の名称】 データ処理装置の制御方法

(57)【要約】

【課題】 处理速度の異なる複数のデータ処理装置を装備した情報処理システムにおいて、各々の処理要求について要求の発生順と処理の終了順の逆転を抑えること、およびシステム全体の処理時間を短くするようなスケジューリングを行うことである。

【解決手段】 スケジューラの処理において、発生した処理要求について各データベース処理装置における見込処理時間と、各データベース処理装置ごとの処理要求の待ち行列全体の処理時間と、各データベース処理装置において処理中の処理要求の残存時間とをそれぞれ見積るステップ、その残存時間と待ち行列全体の処理時間と見込処理時間とを合計した処理終了見込時間の最も小さいデータ処理装置の待ち行列に発生した処理要求を追加するステップから構成される。



1

## 【特許請求の範囲】

【請求項1】 处理速度の異なる複数のデータ処理装置を備え、複数の処理要求をその要求が発生した順にスケジューリングを行う情報処理システムにおいて、発生した処理要求について上記各データ処理装置における見込処理時間と、上記各データ処理装置ごとの処理要求の待ち行列全体の処理時間と、上記各データ処理装置において処理中の処理要求の残存時間とをそれぞれ見積るステップ、その残存時間と待ち行列全体の処理時間と上記見込処理時間とを合計した処理終了見込時間の最も小さいデータ処理装置の待ち行列に上記発生した処理要求を追加するステップからなるデータ処理装置の制御方法

【請求項2】 上記処理要求について上記各データ処理装置における見込処理時間及び上記各データ処理装置における処理要求の待ち行列全体の処理時間を見積るステップにおいて、上記処理時間を見積るために各処理要求のデータ量と各データ処理装置に対する負荷情報を用いることを特徴とする請求項1に記載のデータ処理装置の制御方法

【請求項3】 上記各データ処理装置における処理要求の待ち行列全体の処理時間を見積るステップにおいて、処理中の各処理要求の経過時間を測定し、その経過時間から各処理中の処理要求の処理終了見込時間を修正することを特徴とする請求項に記載のデータ処理装置の制御方法

【請求項4】 上記処理要求について上記各データ処理装置における見込処理時間及び上記各データ処理装置における処理要求の待ち行列全体の処理時間を見積るステップにおいて、処理要求の発生分布と処理要求の処理時間のそれぞれの統計情報を上記見積りの参考にすることを特徴とする請求項1に記載のデータ処理装置の制御方法

【請求項5】 上記処理要求の発生間隔の平均時間と上記各データ処理装置における処理時間の平均時間を採取し、発生間隔の平均時間が処理時間の平均時間よりも大きいときは上記複数のデータ処理装置間の処理速度の性能比の倍率に発生間隔の平均時間の処理時間の平均時間に対する比率を乗じた倍率を上記複数のデータ処理装置間の処理速度の性能比の倍率として上記各データ処理装置ごとの処理時間を見積ることを特徴とする請求項1に記載のデータ処理装置の制御方法

【請求項6】 上記各データ処理装置ごとの処理時間が見積時間と比較して誤差が大きいとき、上記各データ処理装置ごとの処理要求の待ち行列を全体的に組変えることを特徴とする請求項1～請求項5のいずれかに記載のデータ処理装置の制御方法

## 【発明の詳細な説明】

【0001】

【発明の属する技術分野】 この発明は、処理性能の異なる複数のデータ処理装置を有する情報処理システムにお

2

いて、複数の処理要求を異なるデータ処理装置によって処理する場合のスケジューリングに関するものである。

【0002】

【従来の技術】 複数のデータ処理装置に対して複数の処理要求を振り分ける従来の方法は、例えば特開昭60-237568号公報、特開平4-373063号公報などで公開されている。前者は複数のデータ処理装置に対して現在および将来の負荷予想を立て、相対的に低負荷状態の計算機に出力処理をさせるものであり、各計算機ごとの負荷状態に着目し負荷の均一化を図ることによって効率の良いシステムを実現させようとしたものである。後者は図10のフローチャートに示すように、データ処理装置が空き状態になる度に処理要求を順番にデータ処理装置に振り分けるものであり、各処理要求を1つのキュー上に溜めて、その順番に空きのデータ処理装置を使用することにより、時系列に沿って順次発生し続ける要求の発生順に処理するものである。

【0003】

【発明が解決しようとする課題】 従来新しいシステムを作る際には、あえて性能が異なるデータ処理装置を用いることはなかったが、過去の資産をそのまま継承しながらより高性能なデータ処理装置を追加したり、他システムとの結合や購入価格の問題などで実際には性能の異なるデータ処理装置を同時に利用する機会が多い。複数のデータ処理装置に処理要求を振り分ける従来の方法は主に上記のような処理を行っているため、装備されているデータ処理装置の間で性能に差がある場合、どのデータ処理装置で処理するかによって、速く処理される要求と遅く処理される要求との違いが発生し、処理要求の発生順と終了順の逆転が生じてしまう。処理要求を出すのが異なるユーザである場合、個々のユーザは後続の要求が自分より先に処理されることに不満を感じるであろう。また従来の方法では、性能の良いデータ処理装置と劣るデータ処理装置を同等に扱うため、負荷が少ないからと言って不用意に性能の劣るデータ処理装置を用いると、全体としても処理の効率が悪くなることもしばしばある。必ずしも全てのデータ処理装置を均等に利用する必要はなく、処理要求の順番に対してどの順番でどのデータ処理装置を利用するかが問題となる。

【0004】 この発明は上記のような問題点を解消するためのもので、第1の目的は、時系列に沿って順次発生する各々の処理要求について、要求の発生順と処理の終了順の逆転を抑えることである。また第2の目的は、情報処理システム全体の処理時間を短くするようなスケジューリングを行うことである。

【0005】

【課題を解決するための手段】 この発明に係るデータ処理装置の制御方法は、処理速度の異なる複数のデータ処理装置を備え、複数の処理要求をその要求が発生した順にスケジューリングを行う情報処理システムにおいて、

発生した処理要求について上記各データ処理装置における見込処理時間と、上記各データ処理装置ごとの処理要求の待ち行列全体の処理時間と、上記各データ処理装置において処理中の処理要求の残存時間とをそれぞれ見積るステップ、その残存時間と待ち行列全体の処理時間と上記見込処理時間とを合計した処理終了見込時間の最も小さいデータ処理装置の待ち行列に上記発生した処理要求を追加するステップからなるものである。

【0006】また、上記処理要求について上記各データ処理装置における見込処理時間及び上記各データ処理装置における処理要求の待ち行列全体の処理時間を見積るステップにおいて、上記処理時間を見積るために各処理要求のデータ量と各データ処理装置に対する負荷情報を用いるものである。

【0007】さらに、上記各データ処理装置における処理要求の待ち行列全体の処理時間を見積るステップにおいて、処理中の各処理要求の経過時間を測定し、その経過時間から各処理中の処理要求の処理終了見込時間を修正するものである。

【0008】また、上記処理要求について上記各データ処理装置における見込処理時間及び上記各データ処理装置における処理要求の待ち行列全体の処理時間を見積るステップにおいて、処理要求の発生分布と処理要求の処理時間のそれぞれの統計情報を上記見積りの参考にするものである。

【0009】さらにまた、上記処理要求の発生間隔の平均時間と上記各データ処理装置における処理時間の平均時間を採取し、発生間隔の平均時間が処理時間の平均時間よりも大きいときは上記複数のデータ処理装置間の処理速度の性能比の倍率に発生間隔の平均時間の処理時間の平均時間に対する比率を乗じた倍率を上記複数のデータ処理装置間の処理速度の性能比の倍率として上記各データ処理装置ごとの処理時間を見積るものである。

【0010】また、上記各データ処理装置ごとの処理時間が見積時間と比較して誤差が大きいとき、上記各データ処理装置ごとの処理要求の待ち行列を全体的に組変えるものである。

【0011】

#### 【発明の実施の形態】

実施の形態1. 以下、この発明の実施形態をデータ処理装置の一例としてデータベース処理装置を例に説明する。図1は性能に違いのある複数のデータベース処理装置5に対して、複数の処理要求3を振り分ける情報システム全体の構成図である。処理要求プログラム1がそれぞれ独立して待ち行列のキューラー2に発生した処理要求3を投げる。一つの処理要求プログラム1が複数の要求を投げることもある。キューラー2はシステムの中の共有メモリ8上のバッファであることが多い。スケジューラ4は、処理要求3が発生したときやデータベース処理装置が空き状態になったときなどの適当なタイミングで、キ

ュー上の処理要求3に対して処理時間の予想を立て、それを元に5a、5bのデータベース処理装置に振り分け実行させる。実際に処理要求をデータベース処理装置5に渡すのは、データベース処理装置5を制御する制御プログラム6である。スケジューラ4は、制御プログラム6を起動して、処理要求3の処理を依頼する。データベース処理装置5と制御プログラム6が実行される計算機と実際にデータ保存されている記憶装置9とは、バス(図示せず)を通じて接続されている。5aが処理速度が速いデータベース処理装置、5bが処理速度が遅いデータベース処理装置である。図1では、簡略のためデータベース処理装置が2台の場合を示しているが、3台以上のデータベース処理装置を装備したシステムであってもよい。

【0012】続いて、スケジューラ4の内部のスケジューリングについて、図2の処理フロー図に基づいて説明する。処理要求3が発生したときやデータベース処理装置5が空き状態になったときなどの適当なタイミングで、図2の一連の処理が行われる。キューラー2上の未処理の処理要求3に対して先頭から順番にデータベース処理装置5が空き状態でなくなるまで、S1からS5までの処理を行う。まず、処理終了時間の予想を立てる処理要求3を1つ取り出す(ステップS1)。次に、取り出された処理要求3について、装備されている全てのデータベース処理装置5において処理したときの処理時間の予想を計算する(ステップS2)。このとき、データベース処理装置5の性能比を基に処理時間の予想を立てる(ステップS2b)。性能比はそのまま処理時間の比として計算される。つまり、性能比が2倍のデータベース処理装置は処理時間が2分の1であるものとする。ここで重要なのは、処理時間の予想を絶対的な時間ではなく相対的な比率で求めることである。そして、それぞれのデータベース処理装置5に対しての見積を比較して、最も早く処理が終了するであろうデータベース処理装置を1つ決定する(ステップS3)。決定したデータベース処理装置5が空いていればそのデータベース処理装置5を用いて処理要求3を処理し(ステップS5)、空いていなければその処理要求はペンディングとして、次の処理要求3の見積を行(ステップS1)。処理自体はスケジューラ4とは別の制御プログラム6を起動させ処理を依頼する。ペンディング中の処理要求3は、割り当てられたデータベース処理装置5で処理するものと仮定し見積に加えるが、実際にそのデータベース処理装置5で処理されるか否かはこの時点では未定である。つまり、先に発生した処理要求3でも、終了時間の予測により空いているデータベース処理装置5で即処理するのではなく、高速なデータベース処理装置5の待ちに入ることが特徴となる。そして更に後続の処理要求3の時間予測の際に、待ちに入るよりは敢えて遅いデータベース処理装置5を使用した方が終了時間は早いと予想されるに至

り。

て、待ちの処理要求3よりも先に処理を開始する。また、処理要求3をキュー2に入れたりキュー2から削除するのはスケジューリングとは別の処理要求プログラム1自身が行う。

【0013】上記手順に従い、図1の5aと5bに見合った性能比を基に見積りを行うことで、遅いデータベース処理装置5bを敬遠し、速いデータベース処理装置5aを有効に活用するスケジューリングを行うことができる。このスケジュールを行ったときの効果を図3を基に説明する。性能比を考慮に入れないスケジューリングの場合には、処理要求11aを速いデータベース処理装置5aに割り当てた後、処理要求11bを空いている遅いデータベース処理装置5bに割り当てる。処理要求11aの処理が終わると次に処理要求11cを5aに割り当て実行させる。このようなスケジューリングでは、処理要求11bの終了時間が処理要求11cのそれより遅くなってしまう。

【0014】これに対し、新しいスケジューリングでは、処理要求12aをデータベース処理装置5aに割り当てた後、処理要求12bは5aで実行した方がよいと判断し(図2のステップS3)5aが空くのを待つ。処理要求12cも同様の理由で5aに割り当てる。処理要求12dに至って遅いデータベース処理装置5bの使用が認められ5bに割り当てられ実行される(図2のステップS5)。また、処理要求12aから12dの処理が終わった時点で、再度残りの処理要求について同様の方法でスケジュールを行う。このようなスケジューリングにより、処理要求12a, 12b, 12c, 12d, 12e, 12fの終了時間の逆転は生じなくなる。さら

に、最後の処理要求である処理要求12fの終了時間も従来のスケジュールの11fに比べて早いものとなる。

【0015】ただし、遅いデータベース処理装置5aで先行処理される処理要求の方が、待ちに入る処理要求に比べてはるかに小さいとき、見積り誤差により上記の方法でも処理の逆転が起きることがある。しかしこの場合の逆転は、不当なスケジューリングに起因するのではなく、処理要求自身の処理内容によるものである。小さいが故に先に終わってしまうのである。厳密に処理順を守らせるには、極端に言えば全ての処理要求を速いデータベース処理装置のみで処理させれば済むが、それでは資源の有効利用にならず、全体的に見て非効率的である。処理順の逆転が起きる場合であっても、上記の方法に従えば全体の処理時間は短くて済む。発明の第1の目的を守りつつ、第2の目的を達成している。

【0016】処理要求3はすなわち処理要求プログラム1であり、ひいてはユーザそのものと見ることができ。上記の方法は、各ユーザそれぞれの立場に立って、自分ができる限り早く処理されるにはどの処理装置に並んだらよいか、を決めるときの方法に従ったスケジュールとなっている。ユーザは、自分より後から並んだ別の

ユーザが、自分より先に処理が終わることを不満に思うであろう。上記方法に従えば、自分が不本意なデータベース処理装置5に割り当てられることによっての終了時間の逆転はなくなる。

【0017】結局、複数ある処理要求3の発生順がデータベース処理装置5の割り当てられ方によってその終了順が逆転することがなくなる様にするには、発生した処理要求3について上記各データベース処理装置5について見積りを行った見込処理時間と、上記各データベース処理装置5ごとの処理要求3の待ち行列全体の処理時間と、上記各データベース処理装置5において処理中の処理要求3の残存時間とをそれぞれ見積り、その残存時間と待ち行列全体の処理時間と上記見込処理時間とを合計した処理終了見込時間の最も小さいデータベース処理装置5の待ち行列に上記発生した処理要求3を追加することによって達成される。

【0018】また、複数ある処理要求3の発生順がデータベース処理装置5の割り当てられ方によってその終了順が逆転することは許しても、情報処理システム全体での処理時間を最小にするには、発生した処理要求3について上記各データベース処理装置5における見込処理時間と、上記各データベース処理装置5ごとの処理要求3の待ち行列全体の処理時間と、上記各データベース処理装置5において処理中の処理要求の終了までの残存時間とをそれぞれ見積り、その残存時間と待ち行列全体の処理時間と上記見込処理時間とを合計した処理終了見込時間の平均値に最も近いデータベース処理装置5の待ち行列に上記発生した処理要求3を追加することによって達成される。

【0019】実施の形態2. 上記実施の形態1では、スケジュールの際の見積条件にデータベース処理装置の性能差のみを用いているが、処理要求3の処理時間は各々の処理要求3の処理内容によっても大きく異なる。よって、処理内容を見積り加えたスケジューリングを考えられる。実施の形態2の構成図は、図1と同様である。

【0020】処理内容を考慮に入れたスケジューリングについて、図4の処理フロー図に基づいて説明する。図2の処理フローと同様に、未処理の各処理要求3について全てのデータベース処理装置5に対する見積り求め、40 割り当てるデータベース処理装置5を決定する。図2と異なるところは、処理時間予想を立てるステップ(ステップS12b)で、データベース処理装置の性能差に加え処理要求3の処理内容を判断材料に利用しているところである。処理内容を表すものとして、データ量とデータベース処理装置5に対する負荷情報を用いた。データ量は、処理すべきデータのファイルサイズや、ブロックサイズや、レコード件数などが考えられる。上記負荷情報は、そのデータに固有の値で例えば選択条件の数やデータが保存してある記憶媒体の性能比などが考えられる。これらをデータベース処理装置5の性能比に乗じる

ことで、処理時間の見積とする。

【0021】上記の手順に従うことによって、処理要求3の処理時間の見積をより正確に行うことが可能となり、処理要求3の大きさに起因する処理終了順の逆転をかなり抑えることができ、また、情報処理システム全体での処理時間をより正確に最小にすることができる等実施の形態1の効果を高めることができる。

【0022】実施の形態3、更に正確に見積を立てるために、すでにデータベース処理装置5で処理されている処理中の処理要求3の処理時間についても、見積の対象にすることが考えられる。また、この処理中の処理要求3の見積をどのように立てるかによって、それまでの見積に大きな誤差が発生したときの例外処理を行うことができる。処理中の処理要求の見積には、処理経過時間を使う。

【0023】図5の処理フロー図に基づいて、処理中の処理要求3の経過時間を見積に加えるスケジューリングについて説明する。処理終了時間の予想を立てる図1の処理要求3を1つ取り出す(ステップS21a)。取り出された処理要求3が処理中のものであれば、その処理が始まってから現在までの処理経過時間を基に見積を行う(ステップS21c)。そして、キューラー2上の次の処理要求3を取り出す(ステップS21a)。処理中の要求でなければ、実施形態1と同様のステップを続ける(ステップS22a)。以降、実施形態1と同様である。

【0024】処理経過時間をどのように見積に加えるかによって、異なる効果を得ることができる。例えば、処理中の処理要求3の経過時間に比例して残りの処理時間の見積時間を修正するようにすれば、より正確な処理時間の予測をすることができる。また、実際の処理経過時間がスケジューリングの際に立てた当初の見積値を大きく越えるような異常な状態になったときに、残りの処理予想時間を非常に大きく取ることで、後続の処理要求3が別のデータベース処理装置5に割り当てられ易くすることもできる。つまり、原因不明で異常な状態にあると思われるデータベース処理装置5を待っている後続の処理要求3が、しひれを切らして別のデータベース処理装置5に並んだ、ことを実際に実現させることができる。もちろん、経過時間が減るに連れ残りの処理時間の予測を減らしていく、ある閾値を超えたとき異常な状態とみなして極端に予想時間を増やして後続の処理要求3をシャットすると言った、両方の効果を組み合わせることも可能である。結局、処理時間が見積時間と比較して誤差が大きい異常な状態になったときに、データベース処理装置5ごとの処理要求の待ち行列を情報システムにおいて全体的に組変えることにより、状況の変化にリアルタイムで対応することができる。

【0025】実施の形態4、実施の形態1では、処理要求プログラム1によってキューラー2上に溜められた処理要

求3のみをスケジュールの対象としていた。しかし、処理要求3は予め用意されているものではなく、断続的に発生するものである。そこで、まだ発生していない処理要求も考慮に入れたスケジュールを考えることができる。

【0026】実施の形態4の処理フローを図6を用いて説明する。見積を立てる対象となっている処理要求3がその時点で最後に発生した要求(キューラー上で最後の処理要求)である場合、処理要求3の今後予想される発生分布を見積に加える(ステップS32c)。発生分布がある程度密な場合は、後の処理要求が発生するまでの発生間隔が短いため、キューラー2上で最後の処理要求3の見積はあまり変える必要はない。しかし、発生分布がまばらな場合には、後の処理要求が発生するまでの発生間隔が長いため、時間が無駄になる。そこで、キューラー上で最後の処理要求3に対してデータベース処理装置5の性能比を抑えた見積を行うようすることで、遅いデータベース処理装置5が利用され易くなり、処理要求の発生間隔分の時間を無駄にしないスケジュールを行うことができる。以降のステップは、実施の形態1と同様のフローである(ステップS32d~S35)。

【0027】上記のような方法により、これから発生するであろう処理要求を考慮に入れたスケジューリングを行うことができ、データベース処理装置を有効利用することができる。

【0028】実施の形態5、実施の形態1、実施の形態2、実施の形態3および実施の形態4を任意に組合せた構成が考えられる。図7は、上記の実施の形態1から実施の形態4までを同時に含んだ構成図である。性能に違いのある複数のデータベース処理装置において、断続的に発生し処理内容の異なる複数の処理要求を振り分ける。処理要求プログラム1がそれぞれ独立してキューラー2に処理要求3を断続的に投げる。スケジューラ4が適当なタイミングで、キューラー2上の処理中の処理要求3と処理待ちの処理要求3の処理時間の予想を立て、まだ発生していない処理要求の発生分布を考慮しながら、5a、5bのデータベース処理装置に振り分け実行させる。

【0029】続いて、スケジューラ4内部のスケジューリングについて、図8の処理フロー図に基づいて説明する。処理要求3が発生したときやデータベース処理装置5が空き状態になったときなどの適当なタイミングで、図8の処理が行われる。キューラー2上の処理要求3の先頭から順番にデータベース処理装置5が空き状態でなくなるまで、S41からS45までの処理を行う。まづ、処理終了時間の予想を立てる処理要求3を1つ取り出す(ステップS41a)。取り出された処理要求が処理中のものであれば、その処理が始まってから現在までの処理経過時間を基に見積を行い(ステップS41c)、続けてキューラー2上の次の処理要求3を取り出す(ステップS41a)。処理中の要求でなければ、データベース処

理装置5の選択に移る(ステップS42a)。予想を立てる対象となっている処理要求3がその時点で最後に発生した要求(キュー上で最後の処理要求)である場合、処理要求3の発生分布を元にデータベース処理装置5の性能比を抑える(ステップS42c)。次に、処理要求3の処理内容とデータベース処理装置5の性能比を基に処理時間の予想を立てる(ステップS42d)。それぞれのデータベース処理装置5に対しての見積りを比較して、最も速く処理が終了するであろうデータベース処理装置5を1つ決定する(ステップS43)。決定したデータベース処理装置5が空いていればそのデータベース処理装置5を用いて処理要求3を処理し(ステップS45)、空いていなければその処理要求3はペンディングとして、次の処理要求の見積りを行う(ステップS41a)。処理自体はスケジューラ4とは別の制御プログラム6を起動させ処理を依頼する。ペンディング中の処理要求3は、割り当てられたデータベース処理装置5で処理するものと仮定し見積りに加えるが、実際に処理されるかはこの時点では未定である。また、処理要求3をキュー2に入れたりキュー2から削除するのはスケジューリングとは別の処理要求プログラム1自身が行う。

【0030】上記手順に従うことによって、処理要求3のそれぞれの処理内容とデータベース処理装置5aと5bの性能比を基に見積りを行い、遅いデータベース処理装置5bを敬遠し、速いデータベース処理装置5aを有效地に活用したスケジューリングを行うことができる。このスケジュールによって、処理要求間で要求発生順と処理終了順の逆転を抑えることができる。さらに、処理中の処理要求の経過時間や未発生の処理要求の発生分布を基に、より正確な処理時間の見積りと無駄のないデータベース処理装置5の利用が可能となった。また、それぞれ必要な機能のみを組み合わせることも自在である。

【0031】実施の形態6、実施の形態1と実施の形態2では、スケジュールを行うための現在の状態を見積の材料としている。それに加え、実施の形態3では、現在進行形の処理状況を見積の材料としている。また、実施の形態4では、まだ発生していない未来の処理要求を見積の材料とし、実施の形態5で全てを加味して見積もる手段を説明した。さらには、過去の統計情報を見積の材料に加えることが考えられる。図9は、処理要求3の発生分布と処理要求3の実際の処理時間のそれぞれの統計情報を採取し、スケジューラ4にフィードバックする構成図である。統計処理20が発生間隔をチェックし、統計処理21が処理時間をチェックする。

【0032】統計処理20および統計処理21は、スケジューラ4や処理要求プログラム1とも独立している。処理要求プログラム1がキュー2に処理要求3を投げる際に、統計処理20を必ず通す。統計処理20は、それぞれの処理要求3の発生間隔の統計情報を採取し、結果をスケジューラ4に渡す。スケジューラ4は統計処理2

0からの結果を、実施形態5の発生分布を見積に加えるステップ(図8 S42c)を利用する。また、統計処理21は処理要求3が実際にデータベース処理装置により処理された処理時間の統計情報を採取し、結果をスケジューラ4に渡す。スケジューラ4は統計処理21からの結果を、実施形態5の処理中の処理要求3の経過時間を見積に加えるステップ(図8 S41c)を利用する。

【0033】例えば、統計処理20では、現時点から数えて過去n個の処理要求3の発生間隔の平均時間tを採取し、統計処理21では同様に過去n個の処理要求3の平均処理時間sを採取するものとする。また、実施の形態5の発生分布を見積に加えるステップ(図8 S42c)を利用する値をTとし、実施の形態5の処理中要求の経過時間を評価に加えるステップ(図8 S41c)を利用する値をSとする。そして、 $t < s$ ならば $T = 1$ とし、 $t \geq s$ ならば $T = s/t$ とする。また、Sにはsそのものを与える。この場合、S42cでは接続されているデータベース処理装置5のシステム内で最も処理速度の遅いデータベース処理装置5に対する性能比の倍率にTを乗じることにより、その性能比を縮小して処理時間の見積り立てることができる。S41cにおいて処理予想時間に対しては $S / (S + \text{処理経過時間の増加関数})$ を乗じることで処理時間の経過とともに見積時間が減少し、リアルタイムにシステムの稼働状態に対応したスケジュールを行うことが可能となる。上記増加関数としては、例えば一次関数が考えられる。もちろん、nの値を自由に変えるのみならず、日付ごとや曜日ごとや時間帯によってシステム稼働状態が決まっているものに対しては、採取する統計情報を日付単位や曜日単位や時間帯で区切りスケジューラ4に値を渡すようにすることも可能である。

【0034】上記方法により、処理要求3の発生分布および処理時間の統計情報を利用し、データベース処理装置5の性能比以外のシステム固有の環境や使用状況に応じて、柔軟にかつ最適にスケジューリングを行うことができる。

【0035】この発明は、データベース処理装置5が他の処理装置、例えばCPUでも、ハードウェアボードでも、プリンターなどの周辺機器でも、ネットワーク接続の計算機でも、またそれらが混在していても適用することが可能である。

【0036】

【発明の効果】この発明は、以上説明したように構成されているので、以下に示すような効果を奏する。

【0037】残存時間と待ち行列全体の処理時間と見込処理時間とを合計した処理終了見込時間の最も小さいデータ処理装置の待ち行列に発生した処理要求を追加するように構成したので、時系列に沿って順次発生する各々の処理要求について、要求の発生順と処理の終了順の逆

転を抑えることができ、かつ情報処理システム全体の処理時間を短くするようなスケジューリングを行うことができる。

【0038】また、処理時間を見積るために各処理要求のデータ量と各データ処理装置に対する負荷情報を用いるように構成したので、より正確な処理時間の見積が得られる。

【0039】さらに、処理中の各処理要求の経過時間を測定し、その経過時間から各処理中の処理要求の処理終了見込時間を修正するように構成したので、より正確な処理時間の見積が得られる。

【0040】また、処理要求の発生分布と処理要求の処理時間のそれぞれの統計情報を上記見積りの参考にするように構成したので、より正確な処理時間の見積が得られる。

【0041】さらにまた、発生間隔の平均時間が処理時間の平均時間よりも大きいときは複数のデータ処理装置間の処理速度の性能比の倍率を縮小して各データ処理装置ごとの処理時間を見積るように構成したので、処理速度の遅いデータベース処理装置が利用され易くなり、処理要求の発生間隔分の時間を無駄にしないスケジュールを行うことができる。

【0042】また、各データ処理装置ごとの処理時間が見積時間と比較して誤差が大きいとき、各データ処理装置ごとの処理要求の待ち行列を全体的に組変えるように\*

\* 構成したので、状況の変化にリアルタイムで対応することができる。

【図面の簡単な説明】

【図1】この発明の実施の形態1、実施の形態2、実施の形態3および実施の形態4を示す構成図である。

【図2】この発明の実施の形態1の処理フローを示すフローチャートである。

【図3】この発明の実施の形態1を示すタイミングチャートである。

【図4】この発明の実施形態2の処理フローを表すフローチャートである。

【図5】この発明の実施形態3の処理フローを表すフローチャートである。

【図6】この発明の実施形態4の処理フローを表すフローチャートである。

【図7】この発明の実施形態5を示す構成図である。

【図8】この発明の実施形態5の処理フローを表すフローチャートである。

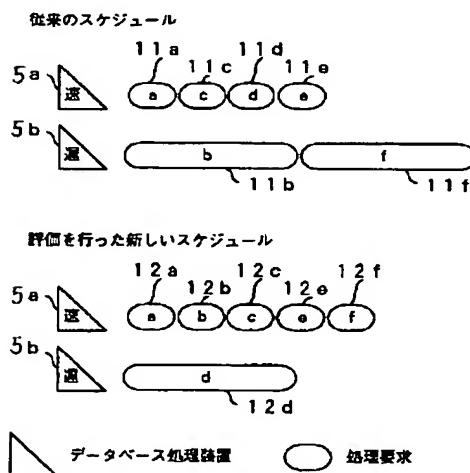
【図9】この発明の実施形態6を示す構成図である。

【図10】従来例のデータ処理装置の制御方法を示すフローチャートである。

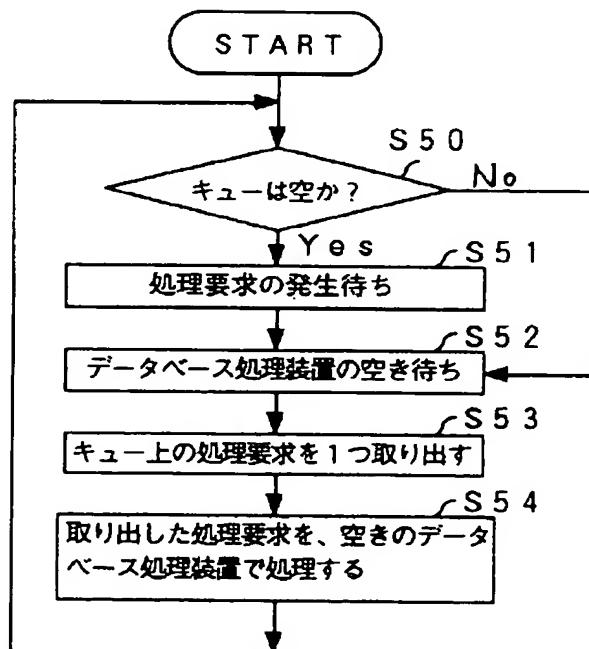
【符号の説明】

3 処理要求、5 データベース処理装置（データ処理装置）、

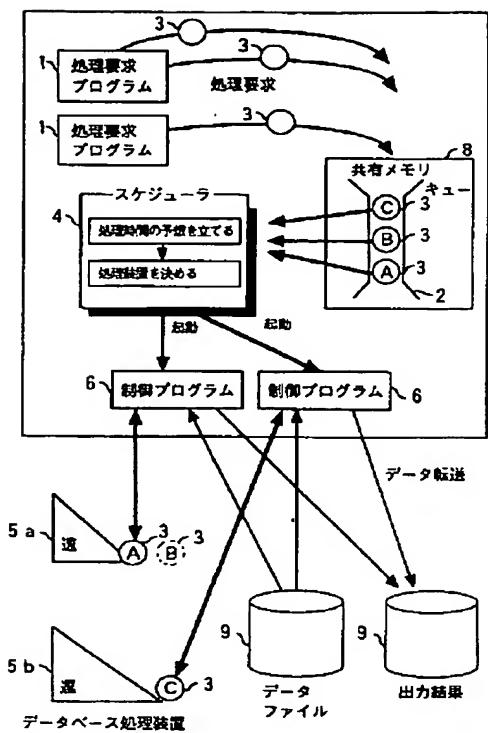
【図3】



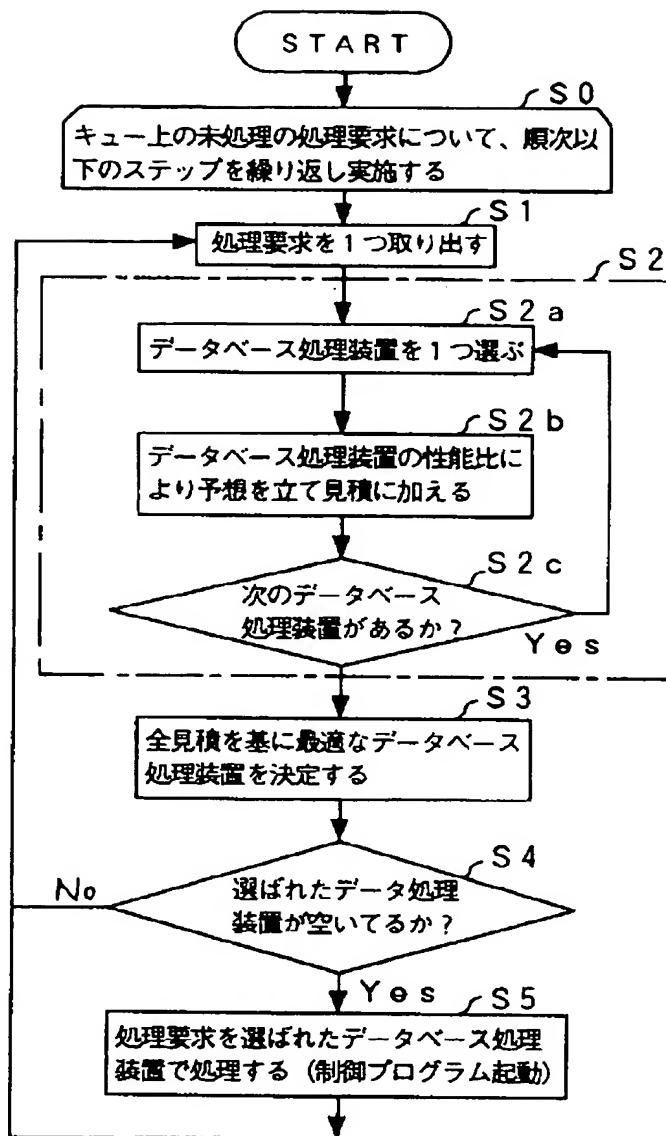
【図10】



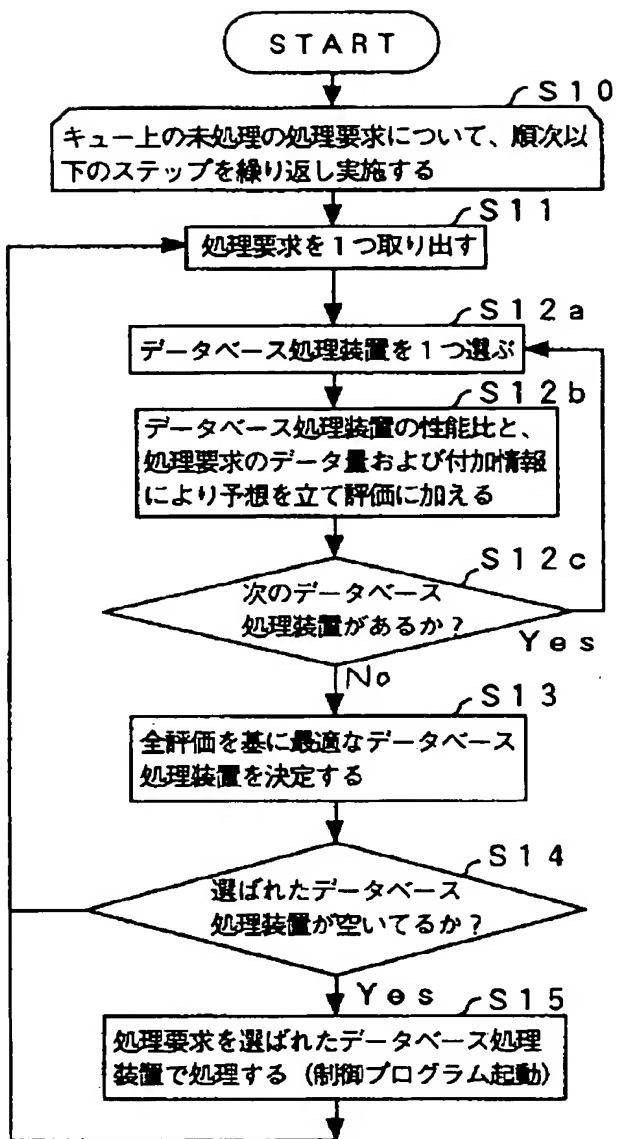
【図1】



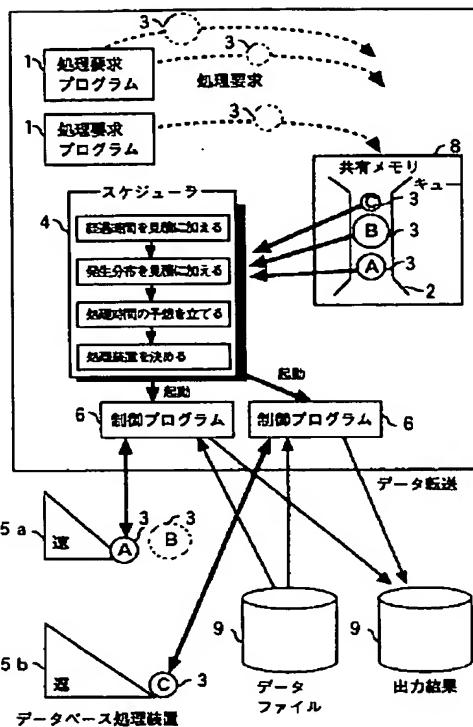
【図2】



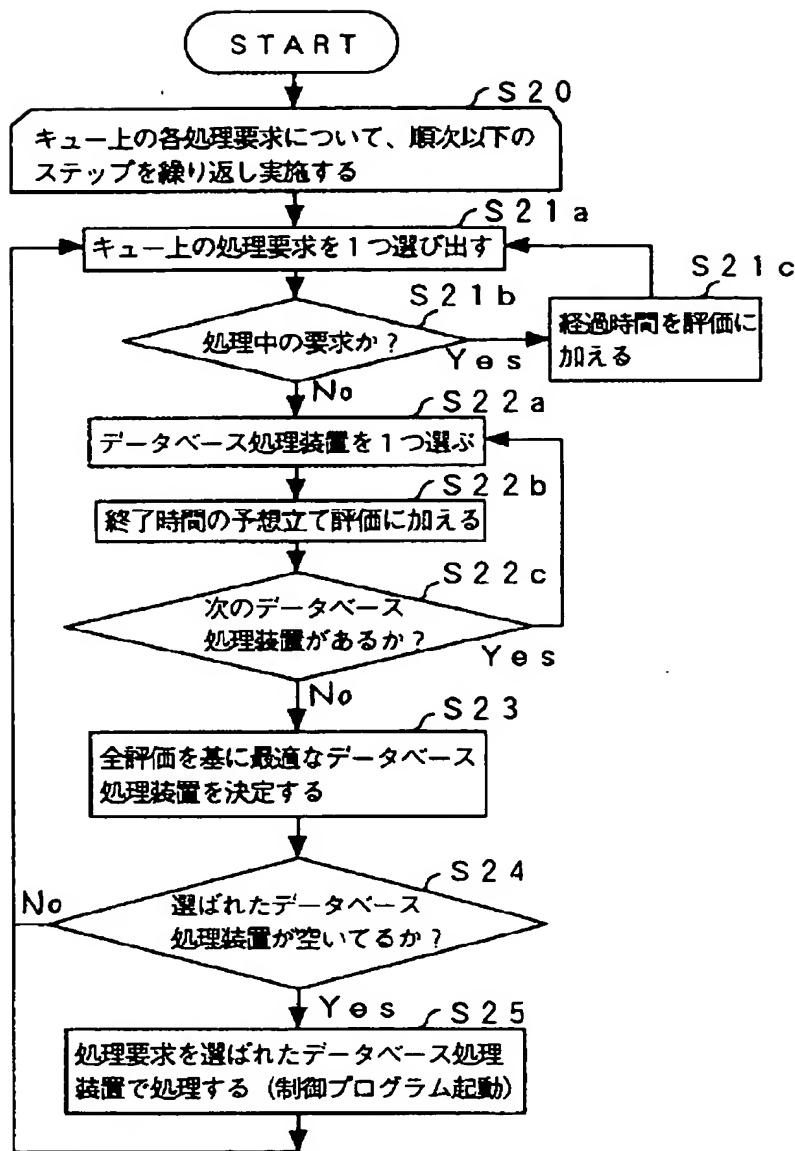
【図4】



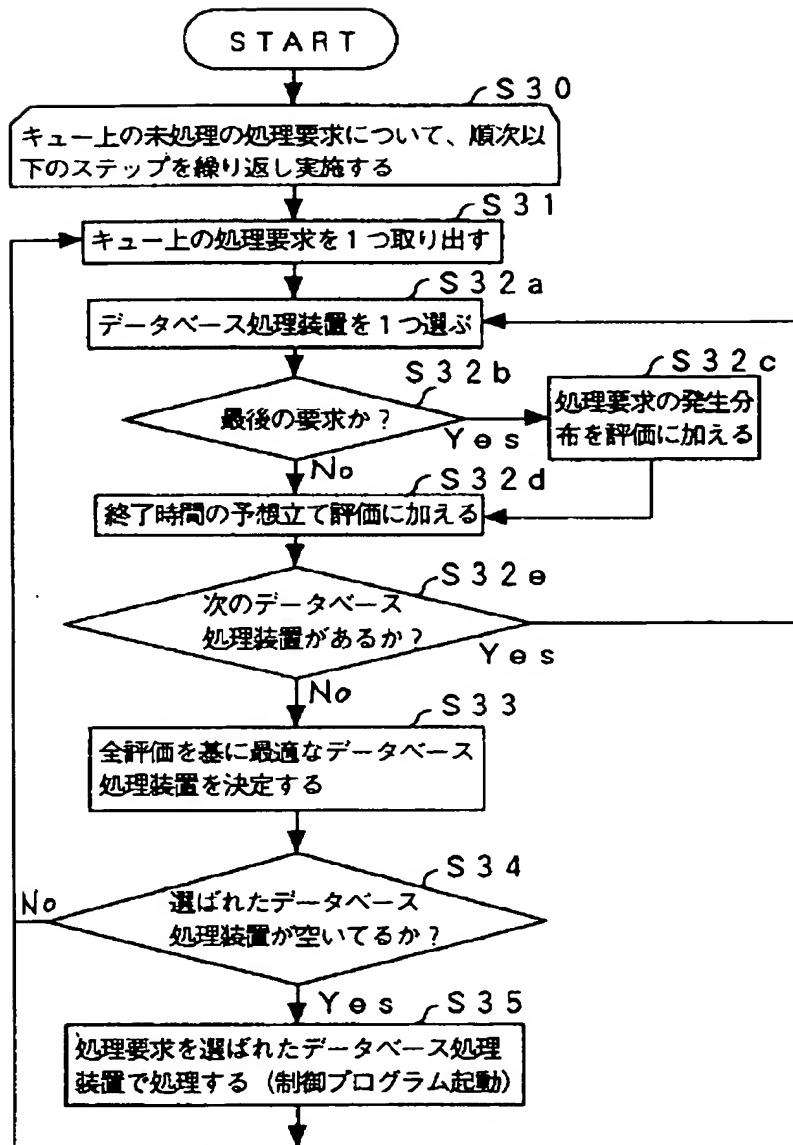
【図7】



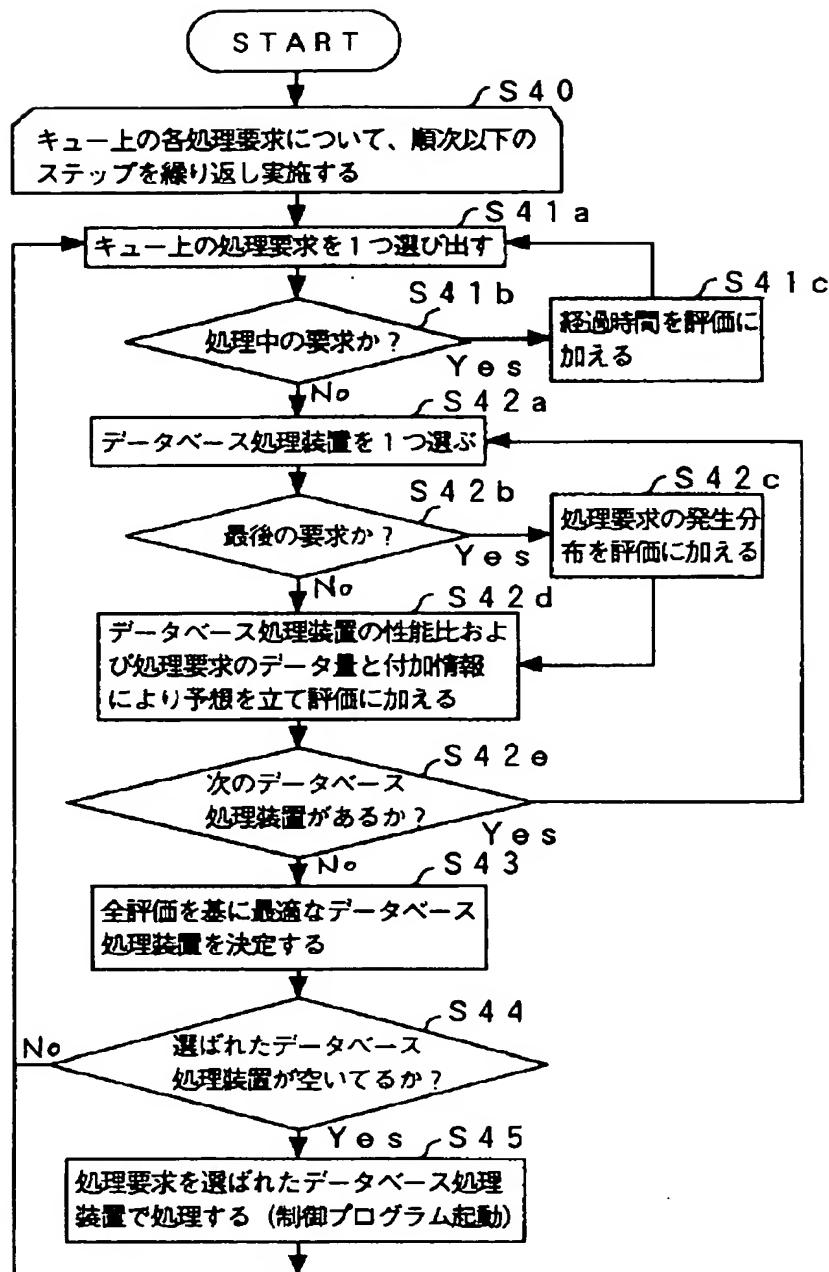
【図5】



【図6】



【図8】



【図9】

